Midterm 1 Review

AW

Lecture Overview

1. **MT composition**

2. **XOR Problem**

3. **Update Multiplying Perceptron**

4. **Update on a small network**

4. True/False and MC questions

5. Computational Graph

Questions and Grading

Composition:

- Undergrad/grad section – very similar to HWs
  - 3 open problem questions
  - T/F-MC section
- Grad only section – not in HWs but in lectures
  - 1 open problem

Grading:

- Grades given in [ ] for undergrad, multiplier fraction for grad students in{ }
- Distribution: 20,20,40 MC/TF 5 each – total 100, max earned 70
- One question has partial credit
  - Updates for the network forward
  - Backward propagation

Lecture Overview

1. MT composition
2. **XOR Problem**
3. Update Multiplying Perceptron
4. Update on a small network

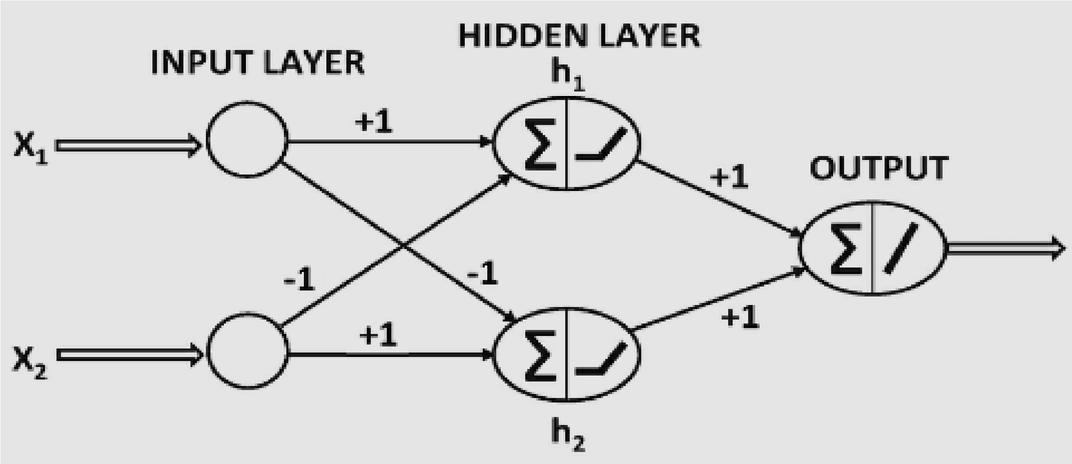
4. True/False and MC questions

5. Computational Graph

XOR problem

- Consider the case of the $\mathbf{XOR}$ function in which the two points $\{(0,0),(1,1)\}$ belong to one class, and the other two points $\{(1,0),(0,1)\}$ belong to the other class. Show how you can use the ReLU activation function to separate the two classes.

Solution to XOR Problem



- Hidden layer contains two ReLU units. Output layer is linear
- Hidden layer should implement the transformations $x_1 - x_2$ and $x_2 - x_1$ to create pre-activation values $\mathrm{W} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$
- For $I = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ we have $M^T I = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- On applying the ReLU activation to the two pre-activated values, one obtains the representation $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- So output is 0 on $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and 1 on $\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ as required

Lecture Overview

1. MT composition

2. XOR Problem

3. **Update Multiplying Perceptron**

4. Update on a small network

4. True/False and MC questions

5. Computational Graph

Update on Multiplying Perceptron Problem

- Consider a two-input neuron that multiplies its two inputs x_1 and x_2 to obtain the output o. Let L be the loss function that is computed at o. Suppose that you know that $\frac{\partial L}{\partial o} = 5$, $x_1 = 2$, and $x_2 = 3$. Compute the values of $\frac{\partial L}{\partial x_1}$ and $\frac{\partial L}{\partial x_2}$.

Multiplying Perceptron Update by BP

- By chain rule $\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial x_1}\bigg|_{\begin{pmatrix}5\\2\\3\end{pmatrix}}$ Since $o = x_1 \cdot x_2$ we have

  $\frac{\partial o}{\partial x_1} = \frac{\partial (x_1 \cdot x_2)}{\partial x_1} = x_2$. Given $\frac{\partial L}{\partial o} = 5$ we obtain

  $$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial x_1}\bigg|_{\begin{pmatrix}5\\2\\3\end{pmatrix}} = 5x_2|_3 = 15.$$

- Similarly $\frac{\partial o}{\partial x_2} = \frac{\partial (x_1 \cdot x_2)}{\partial x_2} = x_1$ so

  $$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial x_2}\bigg|_{\begin{pmatrix}5\\2\\3\end{pmatrix}} = 5x_1|_2 = 10$$

Lecture Overview

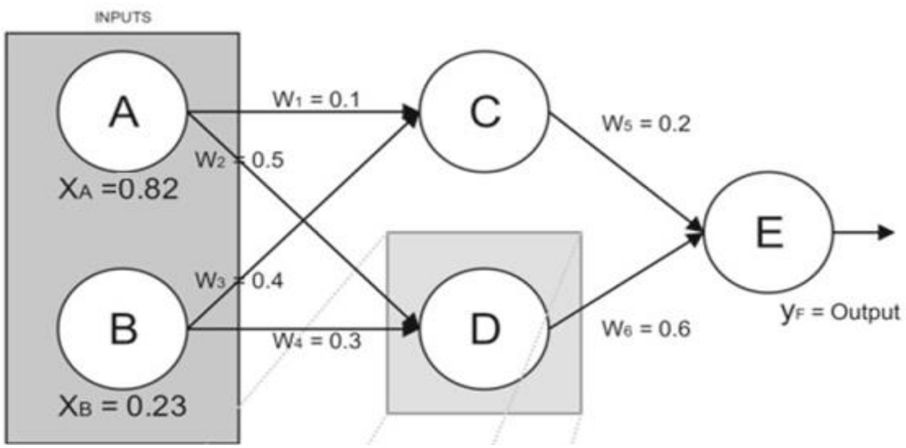
1. **MT composition**
2. **XOR Problem**
3. **Update Multiplying Perceptron**
4. **Update on a small network**

4. True/False and MC questions

5. Computational Graph

Small network update problem

- Suppose that for ANN below we are given training instance $(\vec{x}, y) = \left[\begin{pmatrix}0.82\\0.23\end{pmatrix}, 0\right]$. What would be the updates in this case? Show your computations (i.e. formulas that you are using and which values you are subbing there)



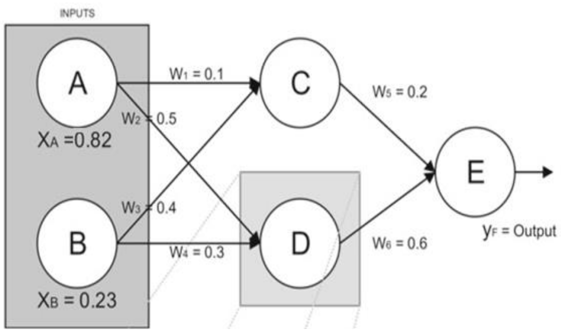
Solution for small network update problem -FP

- Weight vectors are $\vec{w}_C = \begin{pmatrix}0.1\\0.4\end{pmatrix}, \vec{w}_D = \begin{pmatrix}0.5\\0.3\end{pmatrix}, \vec{w}_E = \begin{pmatrix}0.2\\0.6\end{pmatrix}$, so $W_{L1} = \begin{pmatrix}0.1 & 0.5\\0.4 & 0.3\end{pmatrix}$
- Pre-activation of layer L1 is given by the vector $\begin{pmatrix}\hat{y}_c\\\hat{y}_d\end{pmatrix} = \vec{z}_{L1} = \begin{pmatrix}0.1 & 0.5\\0.4 & 0.3\end{pmatrix}^T . \begin{pmatrix}0.82\\0.23\end{pmatrix} = \begin{pmatrix}0.174\\0.479\end{pmatrix}$
- Post-activation values of layer L1 are $\vec{y}_{L1}(\vec{z}_{L1}) = \frac{1}{1+e^{-\vec{z}_{L1}}} = \frac{1}{1+e^{-\begin{pmatrix}0.174\\0.479\end{pmatrix}}} = \begin{pmatrix}0.543\\0.618\end{pmatrix}$
- Pre-activation of layer L2 (=E) is $z_{L2} = \begin{pmatrix}0.2\\0.6\end{pmatrix}^T . \begin{pmatrix}0.543\\0.618\end{pmatrix} = 0.479$
- Post-activation values of layer L2 are $\hat{y}_E = \hat{y}_{L2}(z_{L2}) = \frac{1}{1+e^{-z_{L2}}} = \frac{1}{1+e^{-0.471}} = 0.618$
- Loss is $L = \frac{1}{2}(y - \hat{y}_E)^2 = \frac{1}{2}(0 - 0.618)^2 = 0.191$

Helpful Tables- Gradient of Activation Functions

Activation function $o = \Phi(x)$	**Gradient** $o'_x = \left(\Phi(x)\right)'_x$
Linear (regression) $\Phi(x) = x$	$(x)'_x = 1$
Linear regression on binary targets $\Phi(x) = \mathrm{sign}(x)$	$(\mathrm{sign}(x))'_x = \begin{cases} 0 \text{ everywhere except } x = 0 \\ undefined \text{ at } x = 0 \end{cases}$
Sigmoid $\Phi(x) = \frac{1}{1-\exp(-x)}$	$\left(\frac{1}{1-\exp(-x)}\right)'_x = \frac{\exp(-\mathrm{x})}{(1-\exp(-x))^2} = \Phi(x)(1\text{-}\Phi(x))$
ReLU $\Phi(x) = \max(0, x)$	$(\max(0, x))'_x = \begin{cases} 1 \; if \; x > 0 \\ 0 \; otherwise \end{cases}$
SVM as in linear on binary targets	$(\mathrm{sign}(x))'_x = \begin{cases} 0 \text{ everywhere except } x = 0 \\ undefined \text{ at } x = 0 \end{cases}$
Softmax $[\Phi(x)]_i = \frac{\exp(v_i)}{\sum_{j=1}^{n} \exp(v_j)}$ (later)	$\frac{\partial [\Phi]_i}{\partial \, v_j} = \begin{cases} [\Phi(x)]_i(1 - [\Phi(x)]_i) \; if \; i = j \\ -[\Phi(x)]_j \cdot [\Phi(x)]_i \end{cases}$

Solution for small network update problem -BP



- We need to compute gradient $\nabla_{\vec{w}(f)} = \begin{pmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \frac{\partial L}{\partial w_3} \\ \frac{\partial L}{\partial w_4} \\ \frac{\partial L}{\partial w_5} \\ \frac{\partial L}{\partial w_6} \end{pmatrix}$
- So, level 0 is $\frac{\partial \mathrm{L}}{\partial y_E} = \left(\frac{1}{2}(1 - \hat{y}_E)^2\right)'_{y_E}\Big|_{\hat{y}_x} = -0 + \hat{y}_E|_{0.618} = 0.618$
- The output of level 0 depend on activation z_E of E. Thus, the gradient involves: $\frac{\partial \mathrm{y_E}}{\partial z_E}\Big|_{0.471} = (1 - \hat{y}_E)\hat{y}_E = (1 - 0.618) \cdot 0.618 = 0.236$ and $\delta_E = \frac{\partial \mathrm{L}}{\partial y_E} \cdot \frac{\partial \mathrm{y_E}}{\partial z_E}\Big|_{0.471} = 0.146$

Solution for small network update problem -BP

- Since $z_E = g_1(\vec{w}) + g_2(\vec{w})$ where $g_1(\vec{w}) = w_5 \cdot \hat{y}_C$ and $g_2(\vec{w}) = w_6 \cdot \hat{y}_D$. So
- $\nabla_{\vec{w}(f)} = \delta_E \nabla_{\vec{w}} z_E = \delta_E \mathbb{J}_{\vec{w}}(z_E) \begin{pmatrix} \partial z_E/\partial g_1 \\ \partial z_E/\partial g_2 \end{pmatrix} = \delta_E \left(\mathbb{J}_{\vec{w}}(z_E)\right)^T \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ where
- $\mathbb{J}_{\vec{w}}(z_E) = \begin{pmatrix} (\nabla_{\vec{w}} g_1)^T \\ (\nabla_{\vec{w}} g_2)^T \end{pmatrix}\Bigg|_{\begin{pmatrix} 0.82 \\ 0.23 \end{pmatrix}} = \begin{pmatrix} \nabla_{\vec{w}}(w_5 y_C)^T \\ \nabla_{\vec{w}}(w_6 y_D)^T \end{pmatrix}\Bigg|_{\begin{pmatrix} 0.82 \\ 0.23 \end{pmatrix}}$

$$= \begin{pmatrix} \nabla_{\overrightarrow{w'}}(w_5 y_C)^T & \frac{\partial}{\partial w_5}(w_5 y_C) & \frac{\partial}{\partial w_6}(w_5 y_C) \\ \nabla_{\vec{w}'}(w_6 y_D)^T & \frac{\partial}{\partial w_5}(w_6 y_D) & \frac{\partial}{\partial w_6}(w_6 y_D) \end{pmatrix}\Bigg|_{\begin{pmatrix} 0.82 \\ 0.23 \end{pmatrix}}$$

where $\vec{w}' = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix}$

Solution for small network update problem -BP

$\nabla_{\vec{w}(f)} = \delta_E \left(\mathbb{J}_{\vec{w}}(z_E)\right)^T \begin{pmatrix}1\\1\end{pmatrix}$ where

$$\mathbb{J}_{\vec{w}}(z_E) = \left.\begin{pmatrix} \nabla_{\vec{w'}}(w_5 y_C)^T & \frac{\partial}{\partial w_5}(w_5 y_C) & \frac{\partial}{\partial w_6}(w_5 y_C) \\ \nabla_{\vec{w}'}(w_6 y_D)^T & \frac{\partial}{\partial w_5}(w_6 y_D) & \frac{\partial}{\partial w_6}(w_6 y_D) \end{pmatrix}\right|_{\begin{pmatrix}0.82\\0.23\end{pmatrix}}$$

So, level 1 is: $\left.\frac{\partial}{\partial w_5}(w_5 y_C)\right|_{x_0} = \hat{y}_C$, $\left.\frac{\partial}{\partial w_5}(w_6 y_D)\right|_{x_0} = 0$,

$\left.\frac{\partial}{\partial w_6}(w_6 y_D)\right|_{x_0} = \hat{y}_D$, $\left.\frac{\partial}{\partial w_6}(w_5 y_C)\right|_{x_0} = 0$, thus

$$\mathbb{J}_{\vec{w}}(z_E) = \begin{pmatrix} 0.2 \cdot \nabla_{\vec{w}'}(y_C)^T & \hat{y}_C & 0 \\ 0.6 \cdot \nabla_{\vec{w}'}(y_D)^T & 0 & \hat{y}_D \end{pmatrix}.$$

For the next step we need $\nabla_{\vec{w}'}(w_5 y_C)^T$ and $\nabla_{\vec{w}'}(w_6 y_D)^T$ for which we can now compute prefixes $\delta_C = \frac{\partial y_C}{\partial z_C}$ and $\delta_D = \frac{\partial y_D}{\partial z_D}$, (that is since δ_E is already computed and outside the Jacobian I removed it from these δ's) so we compute $\delta_C = \frac{\partial y_C}{\partial z_C} = (1 - \hat{y}_C)\hat{y}_C = 0.248$ and $\delta_D = \frac{\partial y_D}{\partial z_D} = (1 - \hat{y}_D)\hat{y}_D = 0.236$

Solution for small network update problem -BP

$$\nabla_{\vec{w}(f)} = \delta_E\left(\mathbb{J}_{\vec{w}}(z_E)\right)^T \begin{pmatrix}1\\1\end{pmatrix} \text{ where}$$

$$\mathbb{J}_{\vec{w}}(z_E) = \left.\begin{pmatrix} \nabla_{\overrightarrow{w'}}(w_5 y_C)^T & \frac{\partial}{\partial w_5}(w_5 y_C) & \frac{\partial}{\partial w_6}(w_5 y_C) \\ \nabla_{\vec{w}'}(w_6 y_D)^T & \frac{\partial}{\partial w_5}(w_6 y_D) & \frac{\partial}{\partial w_6}(w_6 y_D) \end{pmatrix}\right|_{\begin{pmatrix}0.82\\0.23\end{pmatrix}}$$

$$= \left.\begin{pmatrix} 0.2 \cdot \nabla_{\vec{w}'}(y_C)^T & \hat{y}_C & 0 \\ 0.6 \cdot \nabla_{\vec{w}'}(y_D)^T & 0 & \hat{y}_D \end{pmatrix}\right|_{\begin{pmatrix}0.82\\0.23\end{pmatrix}}$$

and $\delta_C = (1 - \hat{y}_C)\hat{y}_C = 0.248$ and $\delta_D = (1 - \hat{y}_D)\hat{y}_D = 0.236$

So level 2is

- $\nabla_{\vec{w}'} y_c = 0.248\, \nabla_{\vec{w}'}(z_C) = 0.248\, \nabla_{\vec{w}'}(f_1 + f_2)$ where $f_1 = w_1 x_A$ and $f_2 = w_3 x_B$,

and

- $\nabla_{\vec{w}'} y_D = 0.236\, \nabla_{\vec{w}'}(z_D) = 0.236\, \nabla_{\vec{w}'}(f_3 + f_4)$ where $f_3 = w_2 x_A$ and $f_4 = w_4 x_B$

Solution for small network update problem -BP

$$\nabla_{\vec{w}(f)} = \delta_E \left(\mathbb{J}_{\vec{w}}(z_E) \right)^T \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$ where

$$\mathbb{J}_{\vec{w}}(z_E) = \left. \begin{pmatrix} 0.2 \cdot \nabla_{\vec{w}'}(y_C)^T & \hat{y}_C & 0 \\ 0.6 \cdot \nabla_{\vec{w}'}(y_D)^T & 0 & \hat{y}_D \end{pmatrix} \right|_{\begin{pmatrix} 0.82 \\ 0.23 \end{pmatrix}}$$

So for $f_1 = w_1 x_A, f_2 = w_3 x_B, f_3 = w_2 x_A$ and $f_4 = w_4 x_B$ level 2 is:

$$\nabla_{\vec{w}'} y_C = 0.248\, \nabla_{\vec{w}'}(f_1 + f_2) = 0.248 \begin{pmatrix} (\nabla_{\vec{w}'} f_1)^T \\ (\nabla_{\vec{w}'} f_2)^T \end{pmatrix}^T \begin{pmatrix} \partial z_C / \partial f_1 \\ \partial z_C / \partial f_2 \end{pmatrix}$$

$$= 0.248 \begin{pmatrix} x_A & 0 & 0 & 0 \\ 0 & 0 & x_B & 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.248 x_A & 0 & 0.248 x_B & 0 \end{pmatrix}$$

$$\nabla_{\vec{w}'} y_D = 0.236\, \nabla_{\vec{w}'}(f_3 + f_4) = 0.236 \begin{pmatrix} (\nabla_{\vec{w}'} f_3)^T \\ (\nabla_{\vec{w}'} f_4)^T \end{pmatrix}^T \begin{pmatrix} \partial z_C / \partial f_3 \\ \partial z_C / \partial f_4 \end{pmatrix}$$

$$= 0.236 \begin{pmatrix} 0 & x_A & 0 & 0 \\ 0 & 0 & 0 & x_B \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0.236 x_A & 0 & 0.236 x_B \end{pmatrix}$$

Solution for small network update problem -BP

$\nabla_{\vec{w}(f)}\Big|_{\binom{0.82}{0.23}} = \delta_E\left(\mathbb{J}_{\vec{w}}(z_E)\right)^T \binom{1}{1}\Big|_{\binom{0.82}{0.23}}$ where

$$\mathbb{J}_{\vec{w}}(z_E)| = \begin{pmatrix} 0.2 \cdot \nabla_{\vec{w}'}(y_C)^T & \hat{y}_C & 0 \\ 0.6 \cdot \nabla_{\vec{w}'}(y_D)^T & 0 & \hat{y}_D \end{pmatrix}\Bigg|_{\binom{0.82}{0.23}}$$

$\nabla_{\vec{w}'} y_C = (0.248x_A \quad 0 \quad 0.248x_B \quad 0)$ and $\nabla_{\vec{w}'} y_D = (0 \quad 0.236x_A \quad 0 \quad 0.236x_B)$

Therefore,

$$\mathbb{J}_{\vec{w}}(z_E)|_{\binom{0.82}{0.23}} =$$

$$\begin{pmatrix} 0.2 \cdot 0.248 \cdot x_A & 0 & 0.2 \cdot 0.248 \cdot x_B & 0 & 0.543 & 0 \\ 0 & 0.6 \cdot 0.236 \cdot x_A & 0 & 0.6 \cdot 0.236 \cdot x_A & 0 & 0.617 \end{pmatrix}$$

$$= \begin{pmatrix} 0.041 & 0 & 0.011 & 0 & 0.543 & 0 \\ 0 & 0.116 & 0 & 0.033 & 0 & 0.617 \end{pmatrix}$$

So $\nabla_{\vec{w}(f)} = \delta_E\left(\mathbb{J}_{\vec{w}}(z_E)\right)^T \binom{1}{1}\Big|_{\binom{0.82}{0.23}} =$

$$= 0.146 \begin{pmatrix} 0.041 & 0 & 0.011 & 0 & 0.543 & 0 \\ 0 & 0.116 & 0 & 0.033 & 0 & 0.617 \end{pmatrix}^T \cdot \binom{1}{1}$$

Solution for small network update problem -BP

$$\nabla_{\vec{w}(f)}\Big|_{\binom{0.82}{0.23}} = \delta_E\left(\mathbb{J}_{\vec{w}}(z_E)\right)^T \binom{1}{1}\Big|_{\binom{0.82}{0.23}}$$

$$= 0.146\begin{pmatrix} 0.041 & 0 & 0.011 & 0 & 0.543 & 0 \\ 0 & 0.116 & 0 & 0.033 & 0 & 0.617 \end{pmatrix}^T \cdot \binom{1}{1}$$

$$= \begin{pmatrix} 0.0059 \\ 0.0169 \\ 0.0017 \\ 0.0047 \\ 0.0792 \\ 0.0901 \end{pmatrix}$$

So $\vec{w}' = \vec{w} - 0.7 \cdot \nabla_{\vec{w}}\left(\frac{1}{2}\left(y - y_E(\vec{x},\vec{w})\right)^2\right|_{(\langle(0.82,0.23),0\rangle,0.1,0.5,0.4,.0.3,0.2,0.6)^T}$

$$= \begin{pmatrix} 0.0958 \\ 0.3881 \\ 0.4988 \\ 0.2967 \\ 0.1445 \\ 0.5369 \end{pmatrix}$$

Lecture Overview

1. MT composition

2. XOR Problem

3. Update Multiplying Perceptron

4. Update on a small network

4. True/False and MC questions

5. Computational Graph

T/F and MC

- TF: Given FFNN with standard neurons. Given a training instance reversal of forward computational graph for the FFNN along with the table of derivatives of standard neurons is used to compute backpropagation by dynamic programming

T/F and MC

- Given FFNN with standard neurons. Given a training instance reversal of forward computational graph for the FFNN along with the table of derivatives of standard neurons is used to compute backpropagation by dynamic programming
  - True
- Dimension of a hidden layer is:
  i. The number of incoming connections to the hidden layer
  ii. Number of perceptrons in the hidden layer
  iii. Number of ourgoing connections of from the hidden layer
  iv. None of the above

T/F and MC

- Given FFNN with standard neurons. Given a training instance reversal of forward computational graph for the FFNN along with the table of derivatives of standard neurons is used to compute backpropagation by dynamic programming
  - True
- Dimension of a hidden layer is:

  ii. Number of perceptrons in the hidden layer
- T/F: Learning rate is learned by backpropagation at the time of training

T/F and MC

- Given FFNN with standard neurons. Given a training instance reversal of forward computational graph for the FFNN along with the table of derivatives of standard neurons is used to compute backpropagation by dynamic programming
  - True
- Dimension of a hidden layer is:

  ii. Number of perceptrons in the hidden layer
- T/F: Learning rate is learned by backpropagation at the time of training
  - False
- Regularization is a method of:

  i. Any method that is intended to improve generalization

T/F and MC

- Given FFNN with standard neurons. Given a training instance reversal of forward computational graph for the FFNN along with the table of derivatives of standard neurons is used to compute backpropagation by dynamic programming
  - True
- Dimension of a hidden layer is:

  ii. Number of perceptrons in the hidden layer
- T/F: Learning rate is learned by backpropagation at the time of training
  - False
- Regularization is a method of:
  1. Any method that is intended to improve generalization
  2. Any method of computing backpropagation
  3. Any method of finding initial assignments
  4. All of the above

Lecture Overview

1. MT composition

2. XOR Problem

3. Update Multiplying Perceptron

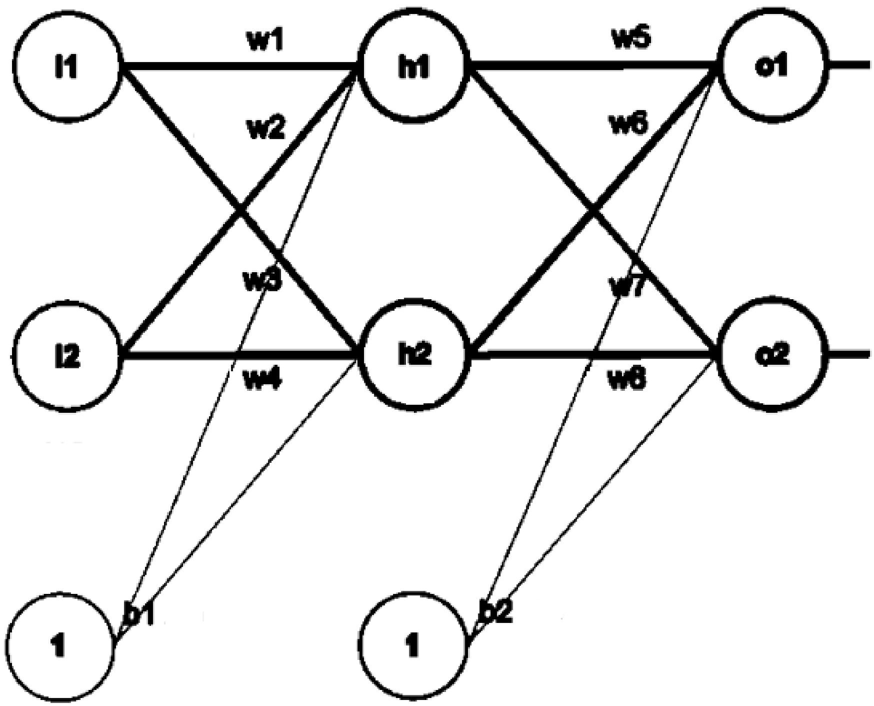
4. Update on a small network

4. True/False and MC questions

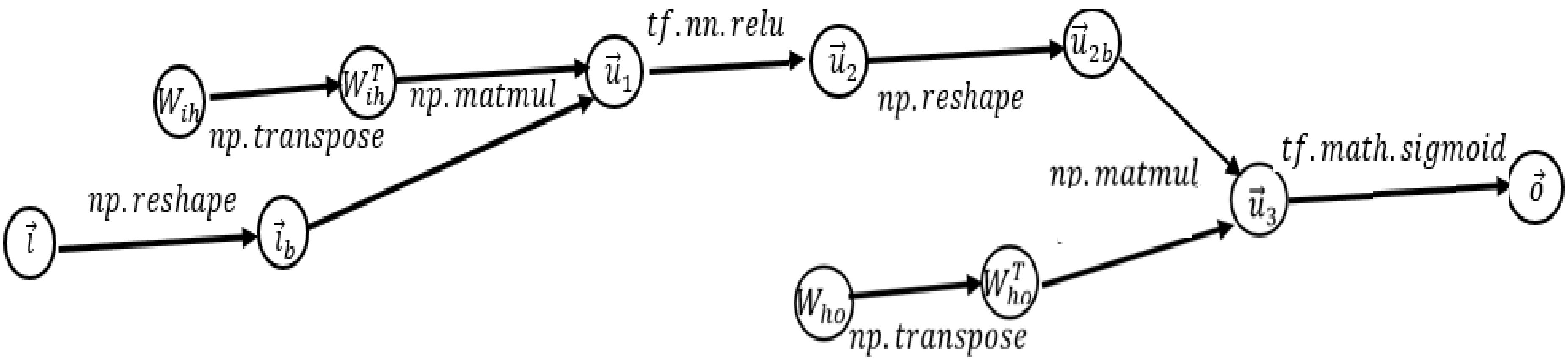
5. Computational Graph

Construct Computational Graph fo NN

- Given the following NN with hidden layer being ReLU and output nodes sigmoids. What is its computational graph assuming loss is SSE?



Computational Graph of the NN



Where:

$$\vec{i} = \begin{pmatrix} I1 \\ I2 \end{pmatrix}, \vec{i}_b = \begin{pmatrix} \vec{i} \\ 1 \end{pmatrix}, W_{ih} = \begin{pmatrix} w_1 & w_3 \\ w_2 & w_4 \\ b_1 & b_1 \end{pmatrix}, \vec{u}_{2b} = \begin{pmatrix} \vec{u}_2 \\ 1 \end{pmatrix},$$

$$W_{ih} = \begin{pmatrix} w_5 & w_7 \\ w_6 & w_8 \\ b_2 & b_2 \end{pmatrix}$$